Universidad
Carlos III de Madrid

Departamento de Teoría de la Señal y Comunicaciones

TRABAJO FIN DE GRADO

Estudio de códigos finitos LDPC y desarrollo de una herramienta simple de diseño

Autor: Carlos Guzmán Velasco

Titulación: Grado en Ingeniería de Sistemas de Comunicaciones

Tutor: Pablo Martínez Olmos

Leganés, Octubre de 2014

Agradecimientos

Aprovecho este pequeño apartado para agradecer a todas aquellas personas que han estado a mi lado apoyándome a lo largo de una de las etapas más importantes de mi vida, todo el proceso de mi formación educativa.

En primer lugar, quiero mencionar a mi madre, mi padre y mi hermano, ya que de una forma u otra siempre me han prestado su apoyo incondicional, tanto en los momentos buenos como en los malos, ayudando así a que estos últimos no lo fueran tanto. Junto a ellos he aprendido muchas cosas que me han hecho crecer como persona y que seguro me servirán en etapas posteriores de mi vida.

En segundo lugar, mencionar a mi novia, Sandra. Aunque no has formado parte de todo lo citado anteriormente, para mí es como si lo hubieras hecho. Desde que te conozco me has apoyado en todo y siempre has estado a mi lado, compartiendo momentos buenos y menos buenos, pero incluso en los menos buenos, siempre conseguías encontrar algo positivo, y eso es algo que admiro en ti. Eres fundamental para mí, ya que sin tu apoyo tampoco sería quien soy ahora y en quien espero convertirme. Espero que estés siempre a mi lado.

También quiero agradecer a mis amigos y al resto de mi familia, ya que en toda etapa educativa siempre se necesitan momentos de esparcimiento y nadie mejor que ellos saben hacer salir de la rutina diaria para despejarte y volver con más fuerzas.

No quiero olvidar a mis compañeros de universidad, con los que estos últimos años he compartido casi más tiempo que con mi propia familia, ya que sin ellos los interminables días, tardes y noches de estudio y prácticas, habrían sido literalmente imposibles de aguantar, porque siempre había alguien que nos arrancaba una sonrisa para olvidar momentáneamente el agobio en el que estábamos inmersos.

Por último, pero no por ello menos importante, quiero mencionar a mi tutor, Pablo, alguien con quien conecté desde la primera clase de prácticas que nos impartió, y a quien tengo que agradecer su interés y dedicación incondicional en todo momento. Sé que con él no me llevo sólo un buen profesor, sino también un gran amigo.

Espero no olvidarme de nadie, por lo que quiero dar las gracias a todas las personas, tanto las que están como las que por desgracia ya no, pero que de alguna manera me han ayudado a ser quien soy. Nunca os olvidaré.

Resumen

"Estudio de códigos finitos LDPC y desarrollo de una herramienta simple de diseño"

Los códigos *LDPC* (*Low-Density Parity-Check*) son los códigos correctores cuyas prestaciones son las que más se acercan a los límites teóricos que estableció Shannon. Debido a esta propiedad, se emplean en una gran variedad de sistemas de comunicaciones.

La característica más importante de estos códigos radica en que sus matrices de chequeo de paridad son de baja densidad, es decir, tienen pocos elementos distintos de 0 y por esta razón, tanto la codificación como la decodificación mediante ellos es muy eficiente en términos de complejidad computacional.

Existen diversos métodos de decodificación para estos códigos, pero el más utilizado es el algoritmo *belief propagation*, también conocido como suma-producto, que se basa en el intercambio de información probabilística estimada para cada bit en cada fila de chequeo de paridad de *H*.

Con este estudio se quiere comprobar que el rendimiento mostrado por códigos *LDPC* cuando la longitud de los códigos tiende a infinito tiene un comportamiento que puede extrapolarse al caso en que los códigos tienen una longitud finita. Para ello estudiaremos estos códigos en un escenario muy sencillo donde podremos extraer un modelo de cómo se comporta el decodificador para códigos finitos.

Abstract

"Finite LDPC codes survey and development of a simple design tool"

The *LDPC* codes (*Low-Density Parity-Check*) are correcting codes whose benefits are those that come closest to the theoretical limits established Shannon. Because of this property, they are used in a variety of communications systems.

The most important characteristic of these codes is that the parity check matrixes are low density, that is, they have few elements different from 0, and for this reason, both encoding and decoding using them is very efficient in terms of computational complexity.

There are several methods of decoding these codes, but the most used is the *belief propagation* algorithm, also known as the sum-product, which is based on the exchange of probabilistic information estimated for each bit in each parity check row from *H*.

With this study we want to verify that the performance shown by LDPC codes when the code length tends to infinity has a behavior that can be extrapolated to the case where the codes have a finite length. We will study these codes in a simple scenario where we can extract a model of how the decoder behaves for finite codes.

Índice

1. Introducción

1.1. Motivación

En los últimos años, los sistemas de comunicaciones que se utilizan en nuestra sociedad van en aumento, y esto junto con el hecho de que cada vez los requisitos de calidad (*Quality of Service, QoS*) de estos sistemas en cuestión de fiabilidad de transmisión son más exigentes, lleva a que los mecanismos usados para la protección y recuperación de los datos transmitidos deban estar en continuo desarrollo.

Al viajar la información a través de canales de transmisión, que en mayor o menor medida afectan y distorsionan siempre su integridad, desde hace años se empezó a investigar para encontrar maneras de proteger dicha información frente al ruido, desvanecimientos… Y aún hoy se investiga en métodos eficientes para alcanzar los límites fundamentales con complejidades asumibles [5][6][7].

Existen muchos métodos para la protección y recuperación de los datos transmitidos, pero en los últimos 15 años se ha impuesto el uso de los códigos *LDPC* (*Low-Density Parity-Check*), ya que se trata de los códigos correctores de errores que mejores resultados han proporcionado hasta la fecha conforme a los límites teóricos establecidos por Shannon.

Estos códigos se usan en infinidad de sistemas de comunicaciones, tales como *DVB-S2* [13], *DVB-T2* [14], *DVB-C2* [15], *DMB-T*, *DMB-H*, *WiMAX* y *WiFi*. También son muy utilizados en sistemas de almacenamiento masivo de información como método para la reconstrucción de la información deteriorada [16].

En este TFG se propone profundizar en técnicas de diseño de códigos LDPC para longitudes finitas. Se han implementado scripts de MATLAB que permiten una primera estimación de la probabilidad de error de un código LDPC en base a los parámetros de un canal sencillo y la longitud del código.

Por esta razón se han elegido estos códigos, con el fin de profundizar en la comprensión de su funcionamiento y estudiar su comportamiento ante distintos escenarios de transmisión.

1.2. Objetivos

Los principales objetivos de este proyecto son:

- Profundizar en el estudio de los códigos bloque LDPC con el fin de conocer mejor su funcionamiento y el de los algoritmos de codificación y decodificación más utilizados en este campo.

- Estudiar el comportamiento de estos códigos en el canal de borrado (*Binary Erasure Channel, BEC*).

- Relacionar el comportamiento de los códigos bloque *LDPC* cuando su longitud tiende a infinito con el comportamiento en el caso finito.

1.3. Glosario de términos

A continuación se listan los acrónimos que aparecen a lo largo del presente documento.

- **AWGN:** Additive White Gaussian Noise.
- **BEC:** Binary Erasure Channel.
- **BER:** Bit Error Rate.
- **BP:** Belief Propagation.
- **BSC:** Binary Symmetric Channel.
- **DVB-C2:** Digital Video Broadcasting – Cable – Second Generation.
- **DMB-H:** Digital Multimedia Broadcast-Handled.
- **DMB-T:** Digital Multimedia Broadcast-Terrestrial.
- **DVB-S2:** Digital Video Broadcasting – Satellite – Second Generation.
- **DVB-T2:** Digital Video Broadcasting – Terrestrial – Second Generation.
- **ETSI:** European Telecommunications Standards Institute.
- **LDPC:** Low-Density Parity-Check.
- **MAP:** Maximum a posteriori.
- **QoS:** Quality of Service.
- **TFG:** Trabajo Fin de Grado.
- **WER:** Word Error Rate.
- **WiFi:** Wireless Fidelity.
- **WiMAX:** Worldwide Interoperability for Microwave Access.

1.4. Estructura de la memoria

A continuación se detallan brevemente los temas tratados en cada uno de los capítulos de la memoria:

- **Capítulo 1:** Se trata de un capítulo introductorio para el lector, en el que están las motivaciones, objetivos y el glosario de términos usados en el documento.

- **Capítulo 2:** Da una visión general de los sistemas de comunicaciones que existen, así como una breve introducción a los métodos de codificación bloque.

- **Capítulo 3:** Da una breve explicación del canal de transmisión que se ha utilizado en este estudio, el canal *BEC*.

- **Capítulo 4:** Presenta una descripción más detallada de los códigos *LDPC* para facilitar así su comprensión.

- **Capítulo 5:** Recoge los resultados obtenidos en las simulaciones, en las que se han ido variando diferentes parámetros de los códigos con el fin de poder modelar el comportamiento de los mismos y se analiza estos resultados.

- **Capítulo 6:** En este capítulo se realiza una estimación de la probabilidad de error a partir de los parámetros obtenidos en el capítulo 5.

- **Capítulo 7:** En él se recopilan los conceptos aprendidos tras la realización del estudio y se plantean posibles líneas futuras sobre el estudio realizado.

2. Estado del arte

2.1. Sistemas de comunicaciones digitales

Los sistemas de comunicaciones digitales emplean técnicas de protección para reducir los errores que se producen cuando la información atraviesa el canal de comunicaciones. Esta protección, consiste en añadir a cada k bits de información r bits redundantes que posteriormente ayuden a detectar y corregir los errores que se producen en la transmisión. El decodificador usará los $n = k + r$ bits recibidos para detectar cuál de las 2^k secuencias posibles fue transmitida y poder recuperar los k bits de información que se enviaron inicialmente.

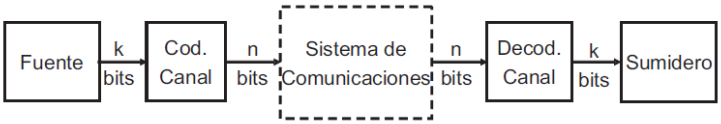

Figura 1. Esquema básico de un sistema de comunicaciones digitales

En la figura 1 se muestra el esquema básico de un sistema de comunicaciones digitales. Al diseñar un codificador, hay que elegir 2^k secuencias de n bits, denominadas palabras código, de tal forma que los errores que introduce el canal de comunicaciones causen el menor número de decodificaciones erróneas. Un decodificador óptimo escoge como palabra código transmitida la que en menos bits difiere de la secuencia recibida, por lo que para minimizar la probabilidad de error las 2^k palabras código deben diferenciarse entre ellas en el mayor número posible de bits.

Se denomina como tasa de transmisión del canal a $R = \frac{k}{n}$. La tasa máxima a la que se puede transmitir por un determinado canal de comunicaciones cumpliendo el requisito de minimizar la probabilidad de error a la salida del decodificador, se conoce como capacidad del canal [1], es decir:

$$C = máx\left(\frac{k}{n}\right) \rightarrow P_e \longrightarrow 0 \qquad (1)$$

2.1.1. Códigos bloque

En la codificación bloque los datos se segmentan en bloques de *k* bits, donde *k* es la longitud de bloque, y el codificador de canal se encarga de transformar cada bloque de *k* bits (añadiendo $n-k$ bits redundantes de una forma predeterminada) en un bloque mayor de *n* bits, donde $n > k$. Cada bloque de *n* bits que se genera tras la codificación se denomina *palabra código*, y al conjunto de 2^k palabras código que conforman un código bloque se les denomina *código de canal*.

La principal característica de los códigos bloque es que cada bloque de *n* bits se genera de forma independiente. Son códigos lineales con estructura de subespacio vectorial, ya que tienen matriz generadora y matriz de chequeo de paridad [2].

Dentro de los códigos bloque, los más utilizados son los lineales, también conocidos como códigos de chequeo de paridad. Estos códigos obtienen las palabras código mediante la suma en módulo dos de subconjuntos de los bits de información a transmitir, por lo que quedan completamente caracterizados su *matriz generadora G*.

2.1.2. Códigos LDPC

Los códigos *LDPC* son códigos bloque lineales cuya característica principal se puede observar en su matriz de chequeo de paridad, que contiene un número de unos muy pequeño en comparación con el de ceros. Cuando el grado de las columnas, es decir, el número de unos por columna es el mismo en toda la matriz, estos códigos se denominan códigos *LDPC* regulares, en caso contrario es denominan códigos *LDPC* irregulares.

La codificación y decodificación de este tipo se lleva a cabo a partir de la matriz *H* (de chequeo de paridad) en la que las filas representan ecuaciones de paridad y las columnas representan símbolos del bloque. Por lo que cada entrada de la matriz cuyo valor sea "1" indica que el símbolo i-ésimo participa en la ecuación j-éisma.

Gallager [3], en 1960, fue el primero en introducir este tipo de códigos, pero debido al elevado coste computacional que suponía su procesamiento y a las limitaciones técnicas de la época quedaron en el olvido hasta años después.

En los últimos años y gracias a la ayuda de los métodos de codificación iterativos se han hecho grandes avances referentes a los códigos *LDPC* [17].

3. Canal BEC

El canal *BEC* (*Binary Erasure Channel*), también conocido como *Canal Binario de Borrado*, se trata de uno de los canales más utilizados a día de hoy en teoría de codificación y en teoría de la información. Fue introducido por Peter Elias del MIT en 1954.

Se trata de un canal que tiene un alfabeto binario (de dos símbolos, 0 y 1). Por lo tanto el transmisor enviará uno de estos dos bits, mientras que el receptor sólo puede recibir el bit que ha enviado el transmisor o un mensaje indicando que no se ha recibido, pero en ningún caso se recibirá un bit erróneo. La probabilidad con la que se borra un bit transmitido se denomina ϵ, por lo que la capacidad del canal será:

$$C = 1 - \epsilon \qquad (2)$$

Con lo explicado anteriormente, se puede modelar este canal según el siguiente esquema.

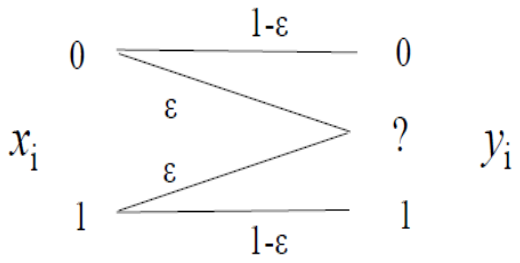

Figura 2. Modelo discreto del canal BEC

Como se puede ver en la Figura 2, este canal está libre de errores, por lo que, a diferencia de otros canales simétricos como el *BSC* (*Binary Symetric Channel*), cuando el receptor recibe un bit, sabe con seguridad que el bit es correcto.

Al realizar este estudio, se ha elegido este canal para realizar las transmisiones por la simplicidad de su implementación y porque gran parte de los canales de transmisión reales pueden modelarse como un canal *BEC*.

3.1. Decodificador óptimo para el canal BEC

Con el fin de minimizar la probabilidad de error de bloque, la decodificación se realiza mediante la aplicación del algoritmo *MAP* (*Maximum a posteriori*).

$$\hat{x}^{MAP} = \arg max_{x \in \mathcal{C}} P(x|y) = \arg max_{x \in \mathcal{C}} P(y|x) \quad (3)$$

En caso de que el conjunto (no vacío) $X(y)$ de palabras código compatible con y contiene sólo una palabra código x, entonces:

$$\hat{x}^{MAP} = x \quad (4)$$

En caso contrario, no existe una solución única.

La complejidad de la decodificación mediante este método depende la longitud de los bloques de código.

Si consideramos que $E \subseteq \{1, \dots, n\}$ determina las posiciones de los bits borrados en y, y F determina las posiciones de los bits recibidos, entonces:

$$x = [x_E x_F] \quad (5)$$

$$xH^T = 0 \Longrightarrow x_E H_E^T + x_F H_F^T = 0 \Longrightarrow x_E H_E^T = y_F H_F^T = s \quad (6)$$

La decodificación MAP puede realizarse resolviendo un conjunto de ecuaciones lineales, es decir, triangularizar mediante el método de Gauss la matriz H_E^T, lo cual tiene una complejidad de $\mathcal{O}(n^3)$. El número de bits borrados es de aproximadamente ϵn y esto hace que para bloques de código de gran longitud, este método sea prohibitivo.

A continuación se presenta un ejemplo gráfico para el método de decodificación óptimo. En este caso se usa un código Hamming (7, 4) y se transmite a través del canal *BEC*.

Este método trata de resolver un bit borrado en cada paso de la decodificación, como se muestra y explica en el siguiente ejemplo.

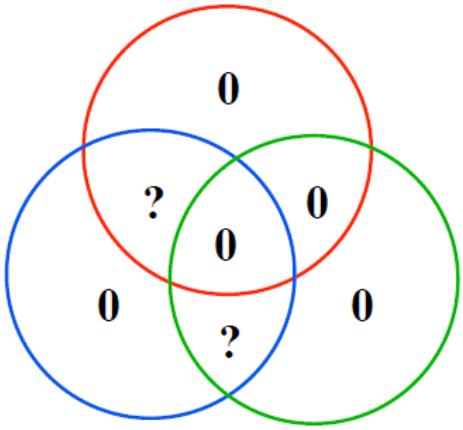

Figura 3. Transmitida la palabra todo 0's con dos borrados

Suponemos que se transmite la palabra todo 0's y que al atravesar el canal *BEC* se borran los dos bits representados en la figura anterior con ?. Las condiciones de paridad relativas al círculo azul no nos ayudarían a resolver los dos borrados, por lo que se pueden usar tanto el círculo rojo como el verde. Si nos centramos en el círculo rojo, vemos que el bit borrado debe ser un 0.

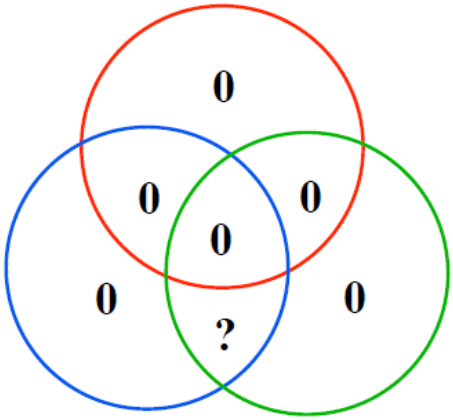

Figura 4. Decodificación tras el primer paso

A continuación repetimos el paso anterior con el bit borrado que falta por decodificar y vemos que, de nuevo, este borrado debe ser un 0. Por lo tanto, la decodificación ha sido correcta ya que la palabra que se había transmitido era todo 0's. Aunque cabe destacar que este procedimiento habría funcionado independientemente de la palabra transmitida.

En capítulos posteriores de este estudio se explicará un método de decodificación subóptimo, el *peeling decoder*, cuya ventaja frente a este método es una menor

4. Códigos LDPC

4.1. Introducción a los códigos LDPC

En su artículo publicado en 1948, Shannon estableció los límites teóricos sobre el comportamiento de la codificación de corrección de errores. Tras este artículo, muchos han sido los esquemas de corrección propuestos, pero ninguno de ellos logró acercarse al comportamiento ideal descrito por Shannon, hasta que en 1993, Berrou, Glavieux y Thitimajshima, descubrieron los esquemas de turbo codificación.

Años más tarde, en 1996, Mackay y Neal [4] redescubrieron unos códigos que habían sido introducidos con anterioridad por Gallager en 1960 y que han logrado un comportamiento cercano al ideal. Estos son los códigos más conocidos como *LDPC*, y son los que se van a analizar en este trabajo. Se trata de códigos de bloque lineales construidos a partir del diseño de una matriz *H* de chequeo de paridad de baja densidad, es decir, en el caso binario, una matriz con un número de unos inferior al de ceros presentes en la matriz. Gallager presentó en su tesis doctoral un método iterativo para la decodificación de este tipo de códigos, pero debido a la escasa capacidad de los procesadores de la época, estos códigos quedaron en el olvido hasta 1996.

4.2. Representación de los códigos LDPC

Existen dos formas de representar los códigos *LDPC*. Una de ellas, como ocurre con todos los códigos bloque lineales es la forma matricial, la otra forma es la gráfica. A continuación se describen brevemente ambas.

4.2.1. Representación matricial

Son matrices de dimensión $m \times n$, denominadas matrices de chequeo de paridad, que quedan definidas mediante dos números:

- *v*: representa el número de unos por fila.
- *s*: representa el número de unos por columna.

Para que se cumpla el que estas matrices sean de baja densidad deben ser matrices de grandes dimensiones en las que $s \ll n$ y $v \ll m$.

4.2.2. Representación gráfica: Grafo de Tanner

El *grafo de Tanner* es un grafo bipartito, introducido por Michael Tanner como un método para mostrar una representación gráfica de los códigos *LDPC*, pero también es de gran ayuda para comprender el algoritmo de decodificación de estos códigos.

Los grafos de Tanner están constituidos por dos subconjuntos de nodos: los *nodos símbolo*, d_j que se corresponden con las filas de *H,* y los *nodos de chequeo de paridad*, h_i que se corresponden con las columnas de *H*. Las conexiones sólo pueden darse entre nodos que pertenezcan a subconjuntos distintos y se simbolizan mediante un "1" en la matriz *H*.

$$\mathbf{H} = \begin{array}{c} \begin{matrix} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 \end{matrix} \\ \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \end{array} \begin{matrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{matrix}$$

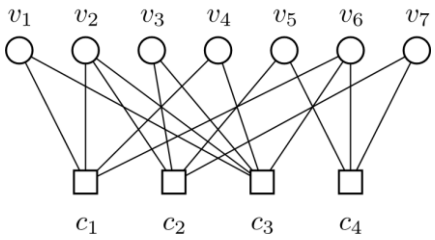

Figura 5. Ejemplo de matriz H de chequeo de paridad con su correspondiente grafo de Tanner

4.3. Formas sistemáticas de los códigos bloque

La matriz generadora G de un código bloque sistemático $c_b(n,k)$ se puede expresar de la siguiente forma:

$$G = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix} = \begin{bmatrix} p_{00} & p_{01} & \cdots & p_{0,n-k-1} & 1 & 0 & 0 & \dots & 0 \\ p_{10} & p_{11} & \cdots & p_{1,n-k-1} & 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k,n-k-1} & 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

Figura 6. Forma sistemática de la matriz generadora, G, de un código bloque

$$G = [PI_k] \quad (7)$$

siendo *P* la submatriz de paridad e I_k la submatriz de identidad ($k \times k$).

El proceso de codificación de la información se realiza mediante la siguiente fórmula:

$$c = G^T \circ m \quad (8)$$

La matriz de chequeo de paridad *H* expresada de forma sistemática es:

$$H = \begin{bmatrix} 1 & 0 & \dots & 0 & p_{00} & p_{10} & \cdots & p_{k-1,0} \\ 0 & 1 & \dots & 0 & p_{01} & p_{11} & \cdots & p_{k-1,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & p_{0,n-k-1} & p_{1,n-k-1} & \cdots & p_{k-1,n-k-1} \end{bmatrix}$$

Figura 7. Forma sistemática de la matriz de chequeo de paridad, H, de un código bloque

$$H = [I_{n-k}P^T] \quad (9)$$

Estas dos matrices deben ser ortogonales, es decir, deben cumplir:

$$H \circ G^T = 0 \quad (10)$$

A partir de *H*, se puede calcular el síndrome, *S*, de un vector de datos codificados, *c*, de la siguiente forma:

$$S = H \circ c \quad (11)$$

4.4. Descripción de los códigos LDPC

La codificación mediante este tipo de códigos se realiza multiplicando el vector de datos a transmitir, *m*, por la matriz generadora, *G*, que caracteriza el código en uso, obteniendo así los datos codificados. La matriz *H* de chequeo de paridad está constituida por vectores fila linealmente independientes que forman a su vez un subespacio dual del generado por los vectores fila de *G*, que son también linealmente independientes. Por lo que para toda palabra código se cumple los siguiente:

$$c \circ H^T = 0 \qquad (2)$$

Los códigos *LDPC* se denotan como $C_{LDPC}(n, s, v)$, donde *n* hace referencia a la longitud del código, *s* indica el número de unos por columna (por lo general $s \geq 3$) y *v* indica el número de unos por fila. La tasa de los códigos *LDPC* se calculará a partir de los tres parámetros anteriores y variará dependiendo de si las filas de *H* son linealmente independientes, en cuyo caso esta tasa será $(v - s)/v$, o no, en cuyo caso la tasa será $(n - s')/n$, siendo *s'* el subespacio vectorial generado por las filas de *H*.

Gallager consiguió demostrar, a partir de la construcción de este código, que la probabilidad de error decrece exponencialmente conforme incrementa la longitud del código y que la distancia mínima aumenta conforme aumenta la longitud del código [17].

4.5. Construcción de los códigos LDPC

4.5.1. Códigos regulares

Gallager propuso un método para construir estos códigos que consistía en crear una matriz *H* poco densa colocando de forma aleatoria los unos, pero siendo el número de estos fijo tanto por columna como por fila.

- La matriz de chequeo de paridad, *H*, debe tener un número *v* fijo de unos por fila.
- La matriz de chequeo de paridad, *H*, debe tener un número *s* fijo de unos por columna.
- El solapamiento de unos por columna y por fila debe ser como máximo igual a uno con el fin de evitar la presencia de ciclos en el correspondiente grafo bipartito. Cabe destacar que esta condición es difícil de cumplir si se quiere que el código sea eficiente.
- Los parámetros *s* y *v* deben ser números pequeños en comparación con la longitud del bloque.

4.5.2. Códigos irregulares

Los códigos irregulares fueron introducidos en [8][9] y posteriormente estudiados en en profundidad en [10][11]. Al contrario que sucedía en los regulares, presentan matrices de chequeo de paridad que no tienen un número fijo de unos por fila y columna. Con estos códigos se obtiene por lo general un mejor resultado en términos de *BER*.

4.6. Decodificación LDPC en el canal BEC

Anteriormente se ha explicado un método de decodificación óptima para códigos *LDPC* sobre el canal *BEC*.

A continuación describimos el método de decodificación, sólo válido para el canal *BEC*, que usaremos en este estudio, el *peeling decoder*. A pesar de ser subóptimo, su complejidad es mucho menor y los códigos *LDPC* están orientados a mejorar las prestaciones del mismo.

4.6.1. Peeling decoder

Otro método de decodificación muy utilizado es el conocido como *peeling decoder*, que consiste en ir borrando de forma iterativa la información que se conoce del grafo de Tanner.

A continuación se muestra un ejemplo que ilustra el funcionamiento de este método.

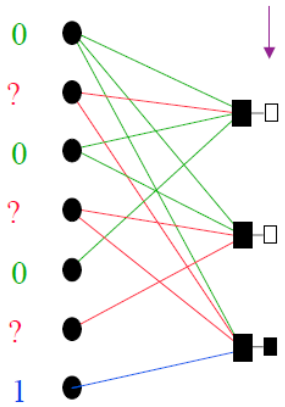

Figura 8. Peeling decoder, paso de inicialización 1

* (**Nota:** *En la figura 8 los recuadros en blanco indican paridad 0*)

En el primer paso de inicialización, cada nodo variable informa de su valor y se comprueba la paridad en los nodos de chequeo.

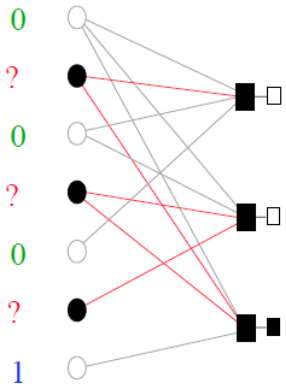

Figura 9. Peeling decoder, paso de inicialización 2

En la siguiente paso se borran del grafo de Tanner las variables que ya se conocen.

Una vez inicializado se procede a la decodificación.

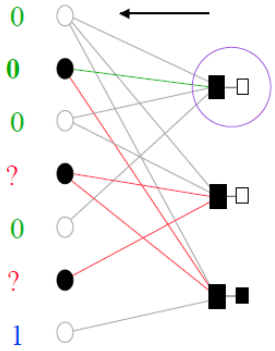

Figura 10. Peeling decoder, decodificación

Se busca un nodo de chequeo de paridad de grado 1 y se borran tanto dicho nodo como el nodo variable que estaba conectado a él. El valor de la variable decodificada es el valor de la paridad del nodo de chequeo en el instante de la decodificación. Por último se debe cambiar la paridad de los nodos de chequeo de paridad conectados a esta variable en caso de que su valor sea 1.

Para concluir se repetirá el proceso explicado anteriormente hasta que todos los nodos variable se hayan borrado, es decir, se ha decodificado correctamente, o hasta que no queden más nodos de chequeo de paridad de grado 1, es decir, no se puede decodificar. En este ejemplo, se decodifica con éxito como se puede ver en las siguientes figuras.

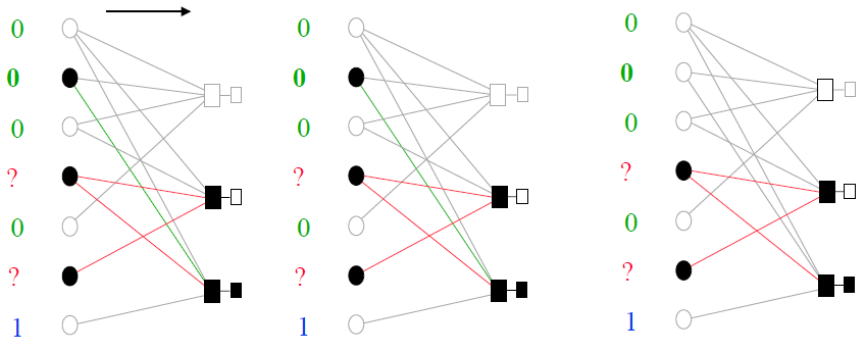

Figura 11. Peeling decoder, proceso de decodificación

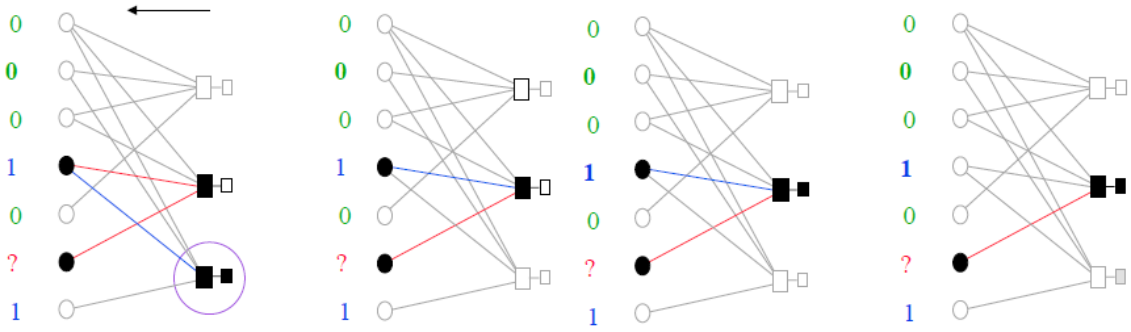

Figura 12. Peeling decoder, proceso de decodificación

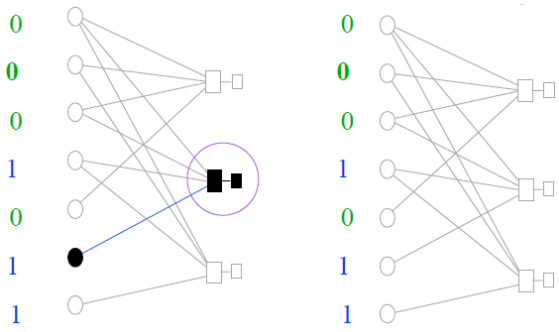

Figura 13. Peeling decoder, decodificación completa

Este proceso puede considerarse como una particularización del *belief propagation* para decodificar códigos bloque lineales en el canal *BEC*.

A pesar de ser fácil de implementar, el *peeling decoder* tiene ciertas desventajas importantes frente a otros métodos de decodificación.

En caso de que no existan nodos de chequeo de paridad de grado 1, es imposible decodificar, como puede verse en la siguiente figura.

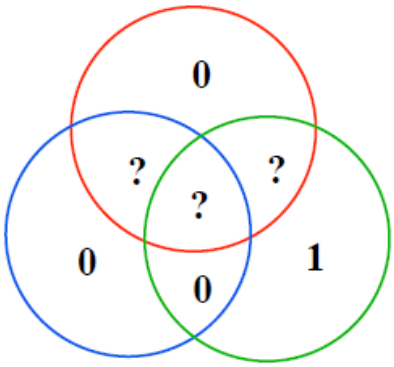

Figura 14. Peeling decoder, todos los nodos de chequeo de paridad de grado>1

Otra de las desventajas de este método se observa cuando en el grafo de Tanner aparecen bucles, en cuyo caso, tampoco se puede decodificar con éxito.

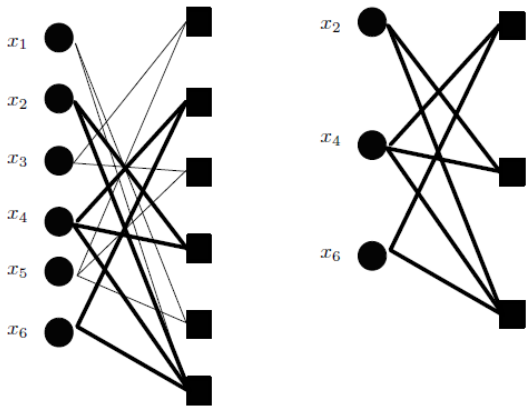

Figura 15. Peeling decoder, bucles en el grafo de Tanner

Como se ha visto en este procedimiento, es muy importante ir haciendo un recuento de los nodos de chequeo de paridad de grado 1 que van quedando según avanza el proceso de decodificación. En adelante denominaremos este proceso estocástico como r_1.

El proceso estocástico será $r_1 = \frac{R_1}{n}$, donde R_1 representa el número nodos de chequeo de paridad de grado 1 que quedan conforme avanza la decodificación. Por lo tanto, r_1 es un proceso normalizado por la longitud del código.

4.6.2. Algoritmo suma-producto

El objetivo de este algoritmo es encontrar un vector decodificado, *d*, como estimación del vector de datos transmitido, *c*.

En el proceso de inicialización de este algoritmo, los nodos símbolo, d_j, envían a cada nodo de chequeo de paridad, h_i, una estimación de que dicho nodo de chequeo de paridad está en un estado. Cada nodo de chequeo de paridad envía a cada nodo símbolo una estimación, calculada a partir de la información recibida de otros nodos símbolo, de que la ecuación de paridad se cumpla en caso de que el nodo símbolo esté en el estado citado anteriormente.

Iterando con este mismo proceso de envío de mensajes, se obtendrá cada vez una mejor estimación de los datos que fueron enviados inicialmente.

Por último recalcar que este algoritmo de decodificación, también conocido como *belief propagation*, se trata de una generalización probabilística del *peeling decoder*.

4.7. Prestaciones para un código regular (3,6)

A continuación se muestran los resultados obtenidos para códigos regulares (3,6) en un canal *BEC* utilizando el método *peeling decoder*.

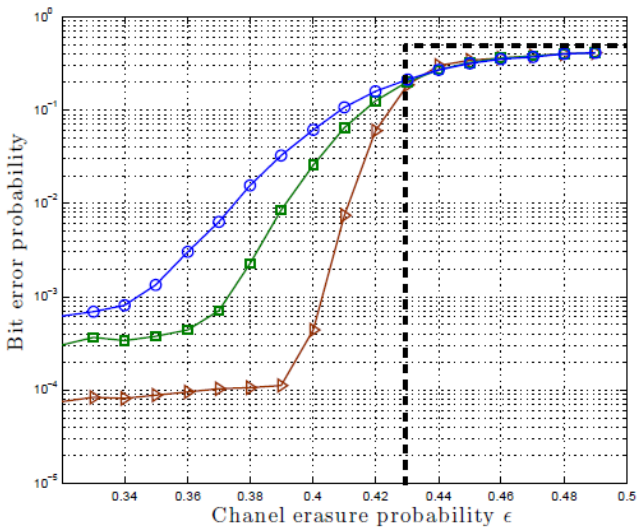

Figura 16. BER para códigos regulares (3,6) de longitudes 2^8 (azul), 2^9 (verde) y 2^{11} (marrón)

Como se puede ver en la figura 16, conforme la longitud de los códigos $n \to \infty$, las 3 gráficas se cortan en un punto, que en adelante denominaremos ϵ_{TH}, en torno al cual, la *BER* puede considerarse despreciable. En el caso de este código el valor de $\epsilon_{TH} = 0.4294$. Este valor de ϵ_{TH} puede calcularse de forma analítica [17].

El objetivo de este proyecto es ver como decae la *BER* hacia la línea punteada conforme aumenta *n*.

5. Simulaciones y resultados

En este estudio buscamos experimentalmente los puntos débiles del código regular (3,6), aunque posteriormente veremos que esto se reproduce en gran cantidad de códigos *LDPC* que se usan hoy día. Una vez localizado, analizaremos el comportamiento en torno a estos puntos con el fin de poder dar una aproximación de la probabilidad de error para cualquier longitud de código, siempre que $\epsilon_{TH_{código}}$ sea conocido.

Los resultados que de $\epsilon_{TH_{código}}$ que usaremos en este estudio se basan en el trabajo de Amaraoui y Urbanke [18].

El objetivo del estudio es el de desarrollar una implementación en MATLAB que permita la obtención de forma rápida y automática de los modelos para distintos códigos con el fin de poder usar esta herramienta con fines didácticos en prácticas de laboratorio.

5.1. Código regular (3,6) de tasa ½

En primer lugar empezamos analizando el código regular (3,6), de tasa ½ y cuyo $\epsilon_{TH_{Reg(3,6)}} = 0.4294$. En la figura 17 se representa la evolución de la media del proceso r_1 para códigos de longitud 2000 y en ella se puede observar con claridad el punto crítico que presenta este código en el proceso de decodificación.

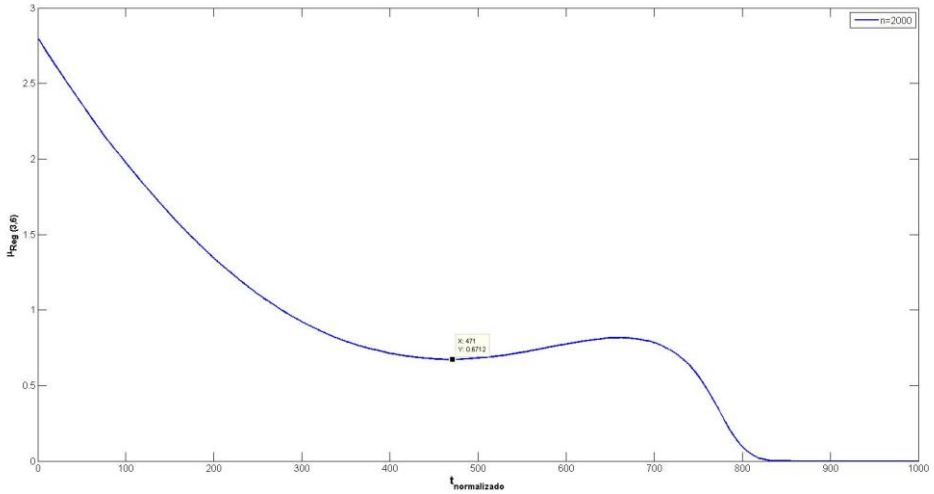

Figura 17. Media del proceso en la que se señala el punto crítico (iteración=471)

* (**Nota:** *En la figura 17, en el eje Y se representa la μ del proceso y en el eje X el tiempo normalizado.*)

Si analizamos el proceso R_1 (recuento del número de nodos de chequeo de paridad de grado 1 que quedan en cada momento) en el punto crítico mediante un histograma (Figura 18), se puede ver que el proceso se comporta aproximadamente (según la Ley de los grandes números) como si fuese gaussiano. Y por lo tanto el proceso estocástico $r_1 = \frac{R_1}{n}$, también tendrá un comportamiento gaussiano.

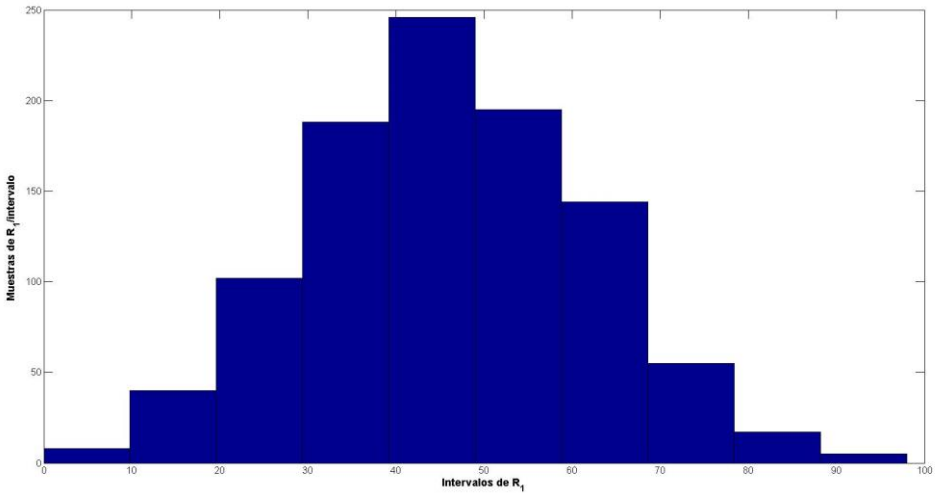

Figura 18. Distribución de los nodos de chequeo de paridad de grado 1 en torno al punto crítico

* (**Nota:** *En la figura 18, en el eje Y se representan el número de muestras de R_1 por intervalo y en el eje X los intervalos de valores de R_1.*)

Se han lanzado simulaciones para códigos de longitud 500, 1000 y 2000, habiendo prefijado la probabilidad de borrado del canal por debajo del umbral de este código ($\epsilon_{TH_{Reg(3,6)}} = 0.4294$) y como se puede ver en la figura 19, las medias para las tres longitudes analizadas confluyen en el punto crítico citado anteriormente. De aquí en adelante normalizaremos el eje de tiempos ya que se están comparando códigos de distintas longitudes.

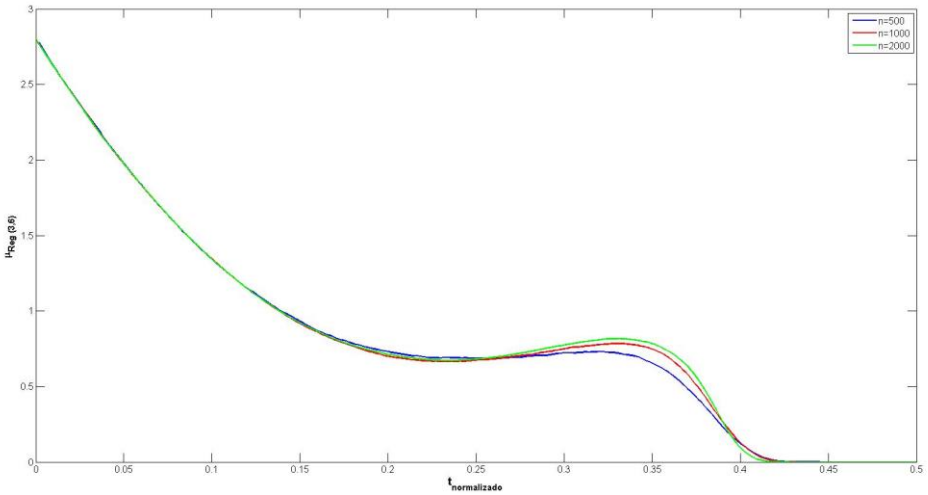

Figura 19. $\mu_{Reg(3,6)}$ para códigos de longitudes 500, 1000 y 2000

* (**Nota:** *En la figura 19, en el eje Y se representan las μ del proceso para distintas longitudes de código y en el eje X el tiempo normalizado.*)

A la vista de la gráfica anterior se puede concluir que la media de r_1 es independiente de la longitud del código, *n*, ya que las gráficas se cortan en el punto crítico independientemente del valor de *n*.

La varianza del proceso r_1, es:

$$\sigma^2 = var(r_1) = \frac{var(R_1)}{n^2} = \frac{var(R_1)}{n} \cdot \frac{1}{n} \qquad (3)$$

donde denotaremos como $v_{código}$ a $\frac{var(R_1)}{n}$. De esta forma, la varianza del proceso queda como:

$$\sigma^2 = var(r_1) = \frac{v_{código}}{n} \qquad (4)$$

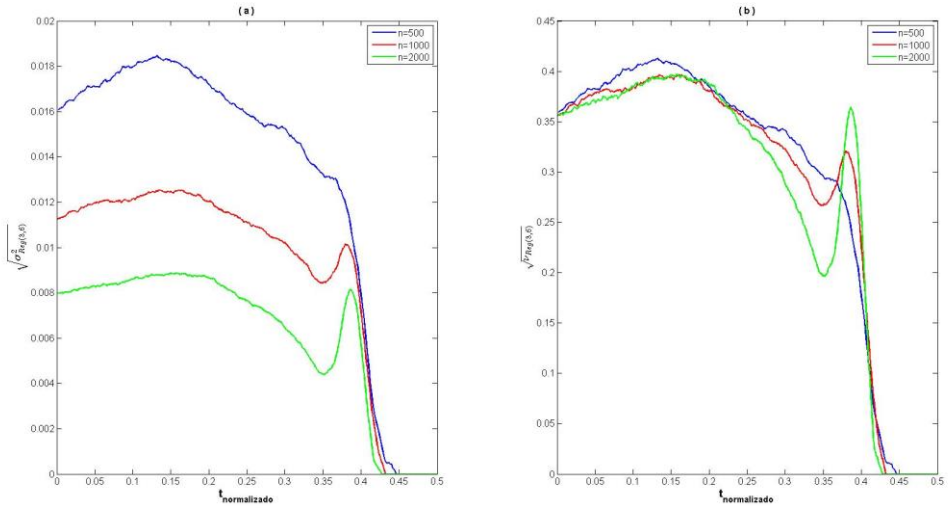

Figura 20. (a) $\sqrt{\sigma^2_{Reg(3,6)}}$ y (b) $\sqrt{v_{Reg(3,6)}}$ para códigos de longitudes 500, 1000 y 2000

* (**Nota:** *En la figura 20 (a), en el eje Y se representa $\sqrt{\sigma^2}$ y en el eje X el tiempo normalizado. En la figura 20 (b), en el eje Y se representa $\sqrt{v_{código}}$ y en el eje X el tiempo normalizado.*)

Las gráficas de la figura 20 ilustran la dependencia de la $v_{código}$ con respecto a la longitud de los códigos utilizados, 'n'. En la gráfica de la izquierda se representa σ^2, mientras que en la gráfica de la derecha se representa $v_{código}$. De esta manera se puede ver que, aunque σ^2 tiene una dependencia cuadrática ($/n^2$) con 'n', las gráficas no muestran esa dependencia ya que la distancia entre las varianzas para códigos de distintas longitudes no es cuadrática. Por esta razón queda demostrada (14). El punto de corte de las gráficas de la derecha nos proporciona el valor de $v_{código}$, que en este caso es:

$$v_{Reg(3,6)} = 0.36^2 = 0.1296$$

A continuación (Figura 21) se ha prefijado la longitud de los códigos a 2000 y se ha ido y se ha ido variando la probabilidad de borrado del canal *BEC*, ϵ, alejándonos cada vez más del umbral. Y aunque se sigue observando que las medias de r_1 siguen teniendo un comportamiento similar al observado en la figura 18, también se aprecia que estas gráficas se van separando levemente unas de otras y tendiendo a 0 conforme decrece ϵ. Por lo tanto se puede concluir que la media del proceso estocástico r_1 es de la forma:

$$\mu \approx \gamma_{código} \cdot \Delta\epsilon \qquad (5)$$

donde:

$$\Delta\epsilon = \epsilon_{TH_{código}} - \epsilon \qquad (6)$$

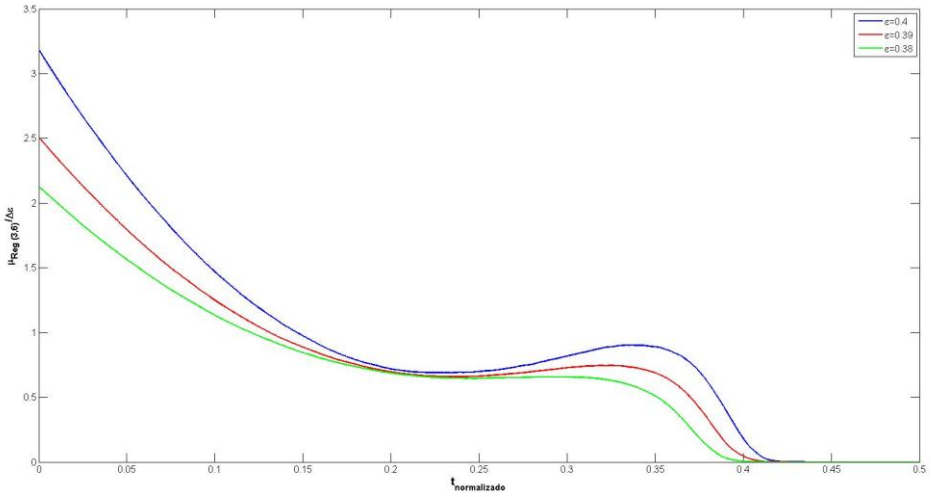

Figura 21. $\mu_{Reg(3,6)}/_{\Delta\epsilon}$ **para** $\epsilon = 0.4$, $\epsilon = 0.39$ **y** $\epsilon = 0.38$

* (**Nota:** *En la figura 21, en el eje Y se representan las* $\mu/_{\Delta\varepsilon}$ *del proceso para distintas ε y en el eje X el tiempo normalizado.*)

En este caso, el punto en el que confluyen las medias nos proporciona el valor de $\gamma_{código}$:

$$\gamma_{Reg(3,6)} = 0.73$$

Por último en la figura 22, y con el mismo procedimiento que en la figura 20 (fijar la longitud del código a 2000 e ir variando ϵ alejándonos cada vez más del umbral), no se observa ningún patrón de comportamiento que denote una dependencia lineal de $v_{código}$ con respecto a $\Delta\epsilon$, por lo tanto la consideraremos despreciable.

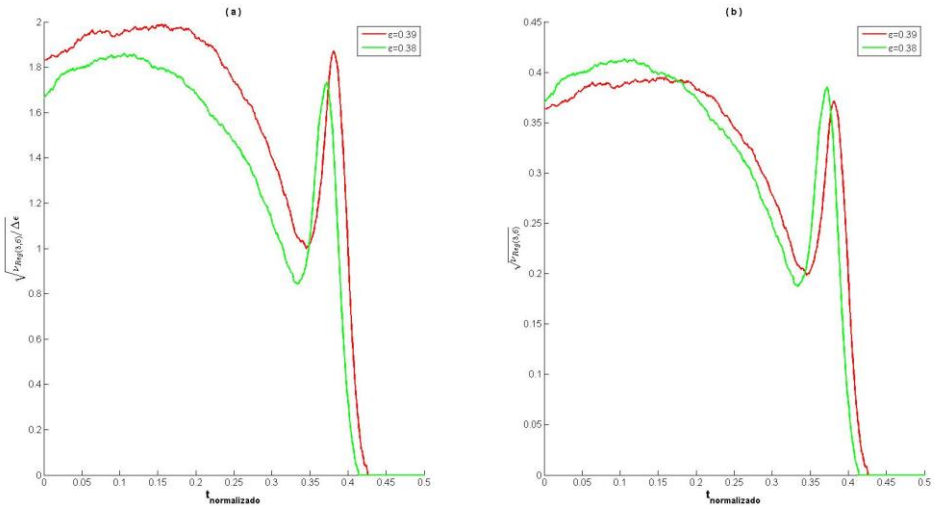

Figura 22. (a) $\sqrt{\nu_{Reg(3,6)}/_{\Delta\epsilon}}$ y (b) $\sqrt{\nu_{Reg(3,6)}}$ para $\epsilon = 0.39$ y $\epsilon = 0.38$

* (**Nota:** *En la figura 22 (a), en el eje Y se representa $\sqrt{\sigma^2/_{\Delta\varepsilon}}$ y en el eje X el tiempo normalizado. En la figura 22 (b), en el eje Y se representa $\sqrt{v_{código}}$ y en el eje X el tiempo normalizado.*)

El comportamiento que hemos visto en este código se reproduce en una gran cantidad de códigos [17].

5.2. Código regular (4,8) de tasa ½

Otro de los códigos con los que hemos probado, el (4,8), es también regular y de tasa ½. Su $\epsilon_{TH_{Reg(4,8)}} = 0.3834$, por lo que presentará unas prestaciones algo peores que el (3,6) como veremos a continuación.

En las figuras 23 y 24, podemos ver que para este tipo de código, se observa el mismo comportamiento en torno al punto crítico que hemos visto en el código regular (3,6).

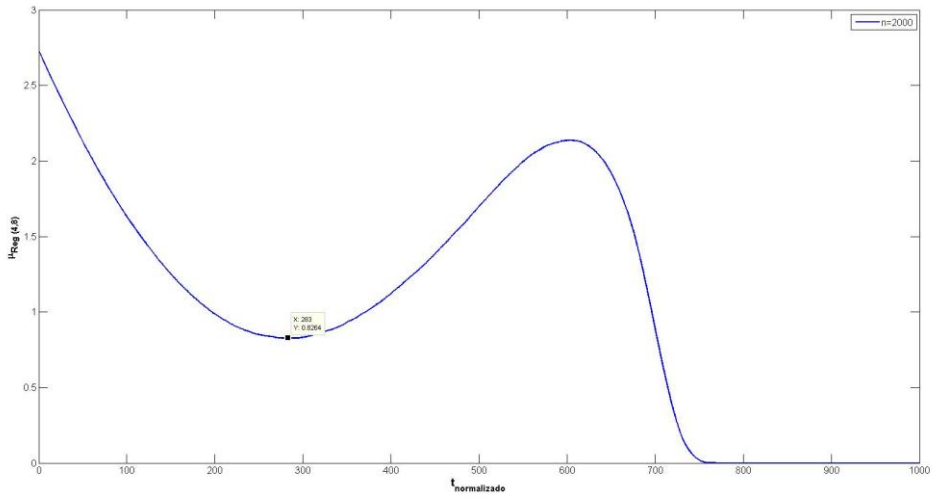

Figura 23. Media del proceso en la que se señala el punto crítico (iteración=283)

* (**Nota:** *En la figura 23, en el eje Y se representa la μ del proceso y en el eje X el tiempo normalizado.*)

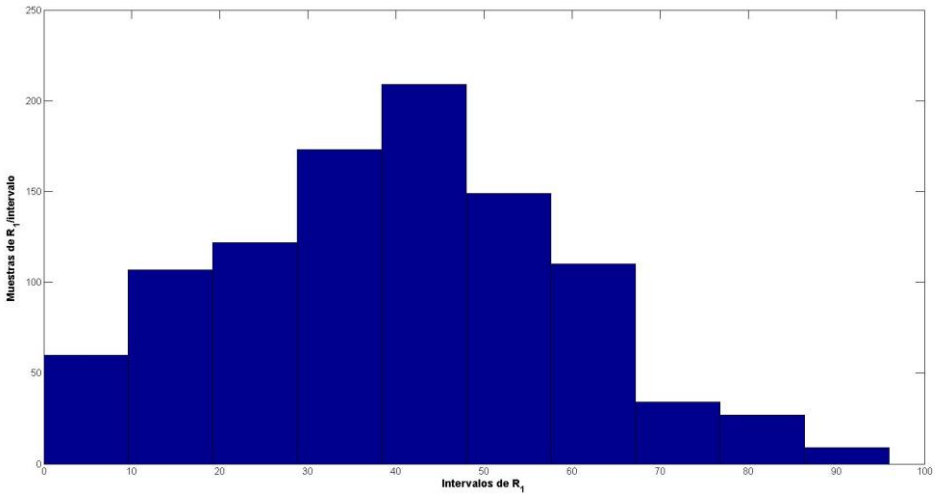

Figura 24. Distribución de los nodos de chequeo de paridad de grado 1 en torno al punto crítico

* (**Nota:** *En la figura 24, en el eje Y se representan el número de muestras de R_1 por intervalo y en el eje X los intervalos de valores de R_1.*)

Al igual que veíamos en la figura 19, las medias siguen confluyendo en el punto crítico y por lo tanto siguen siendo independientes de la longitud del código.

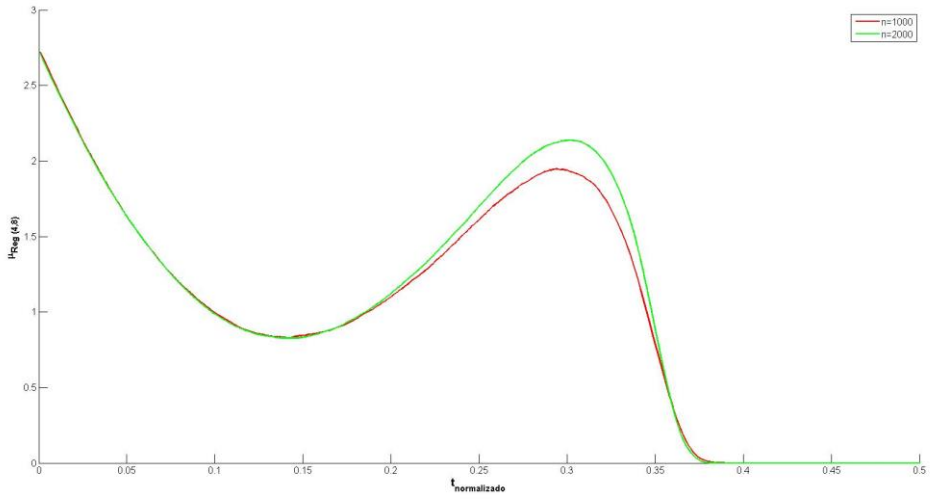

Figura 25. $\mu_{Reg(4,8)}$ para códigos de longitudes 1000 y 2000

* (**Nota:** *En la figura 25, en el eje Y se representan las μ del proceso para distintas longitudes de código y en el eje X el tiempo normalizado.*)

En la figura 26, se vuelve a ver la dependencia $\nu_{código}$ con *n*.

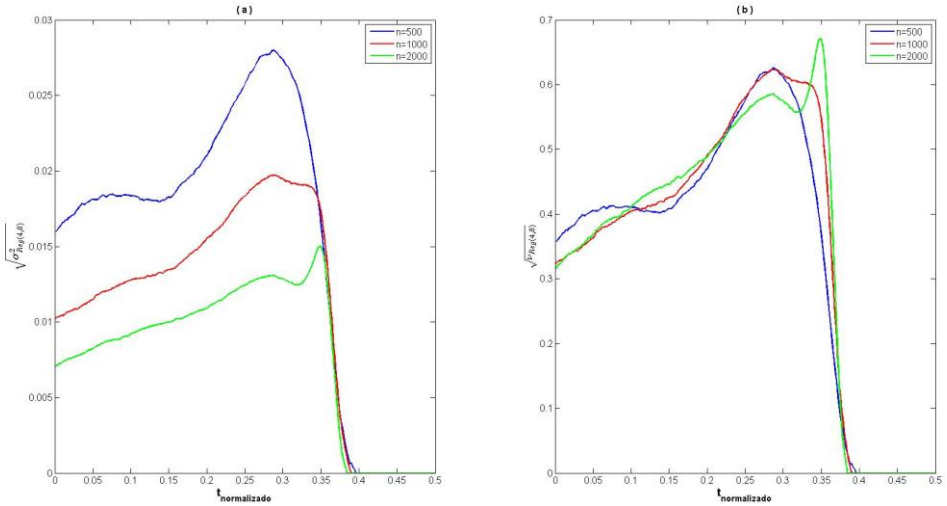

Figura 26. (a) $\sqrt{\sigma^2_{Reg(4,8)}}$ y (b) $\sqrt{\nu_{Reg(4,8)}}$ para códigos de longitudes 500, 1000 y 2000

* (**Nota:** *En la figura 26 (a), en el eje Y se representa $\sqrt{\sigma^2}$ y en el eje X el tiempo normalizado. En la figura 26 (b), en el eje Y se representa $\sqrt{\nu_{código}}$ y en el eje X el tiempo normalizado.*)

En este caso el valor de $\nu_{código}$ es:

$$\nu_{Reg(4,8)} = 0.4^2 = 0.16$$

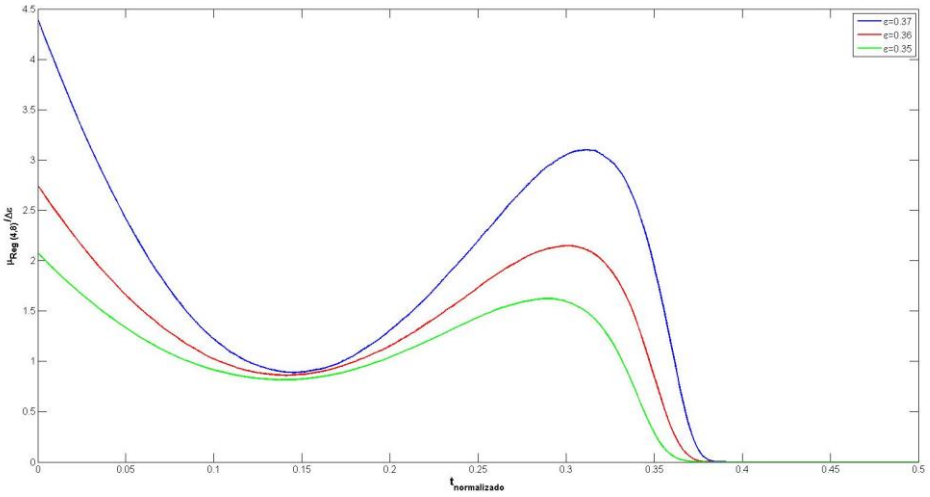

Figura 27. $\mu_{Reg(4,8)}/_{\Delta\epsilon}$ **para** $\epsilon = 0.37$, $\epsilon = 0.36$ **y** $\epsilon = 0.35$

* (**Nota:** *En la figura 27, en el eje Y se representan las* $\mu/_{\Delta\varepsilon}$ *del proceso para distintas* ε *y en el eje X el tiempo normalizado.*)

En la figura 27, volvemos a observar la dependencia de la media de r_1 con respecto a $\Delta\epsilon$. El valor de $\gamma_{código}$ en este caso es:

$$\gamma_{Reg(4,8)} = 0.8164$$

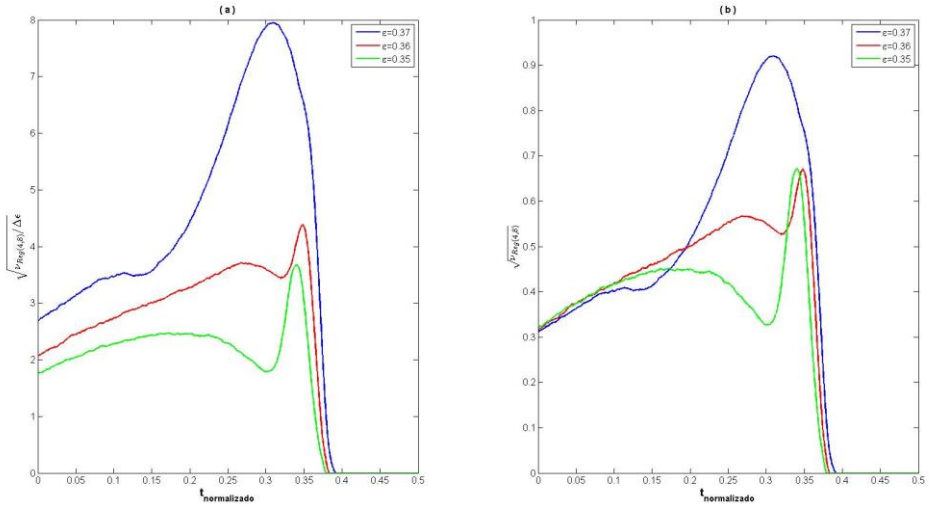

Figura 28. (a) $\sqrt{\nu_{Reg(3,6)}/_{\Delta\epsilon}}$ **y (b)** $\sqrt{\nu_{Reg(3,6)}}$ **para** $\epsilon = 0.37$, $\epsilon = 0.36$ **y** $\epsilon = 0.35$

* (**Nota:** *En la figura 28 (a), en el eje Y se representa* $\sqrt{\sigma^2/_{\Delta\varepsilon}}$ *y en el eje X el tiempo normalizado. En la figura 28 (b), en el eje Y se representa* $\sqrt{\nu_{código}}$ *y en el eje X el tiempo normalizado.*)

Por último, en la figura 28, al igual que ocurría con el código regular (3,6), se puede ver como la dependencia de $\nu_{código}$ con respecto a $\Delta\epsilon$ no es lineal.

5.3. Código irregular optimizado para operar cerca de la capacidad

Para finalizar con este estudio hemos escogido un código irregular [17] que ofrece mejores prestaciones que los dos anteriores ya que su $\epsilon_{TH_{Irr-146Urb}} = 0.482803$.

Al igual que sucedía con el (3,6) y el (4,8), en la figura 29, observamos la existencia del punto crítico del código, en torno al cual el comportamiento del proceso es aproximadamente gaussiano.

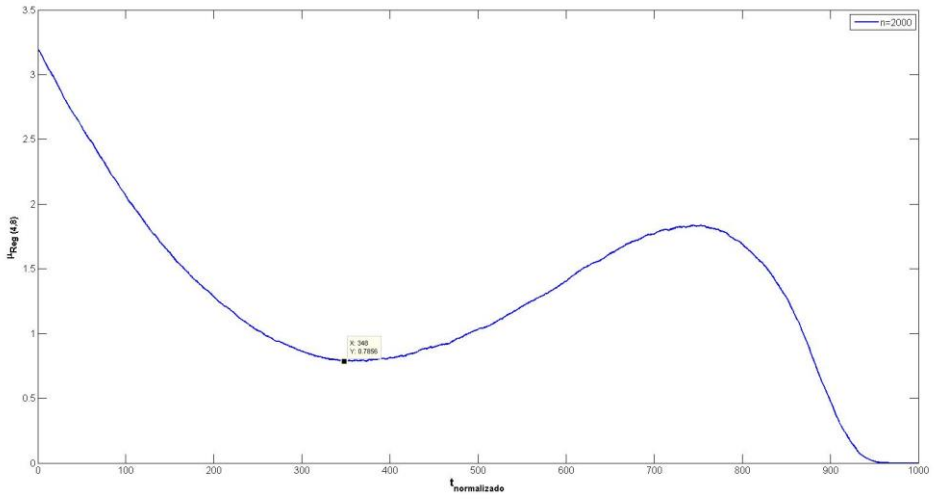

Figura 29. Media del proceso en la que se señala el punto crítico (iteración=348)

* (**Nota:** *En la figura 29, en el eje Y se representa la μ del proceso y en el eje X el tiempo normalizado.*)

En la figura 30, volvemos a observar la independencia de la media con respecto n, ya que las medias vuelven a juntarse en un punto crítico sin importar la longitud del código.

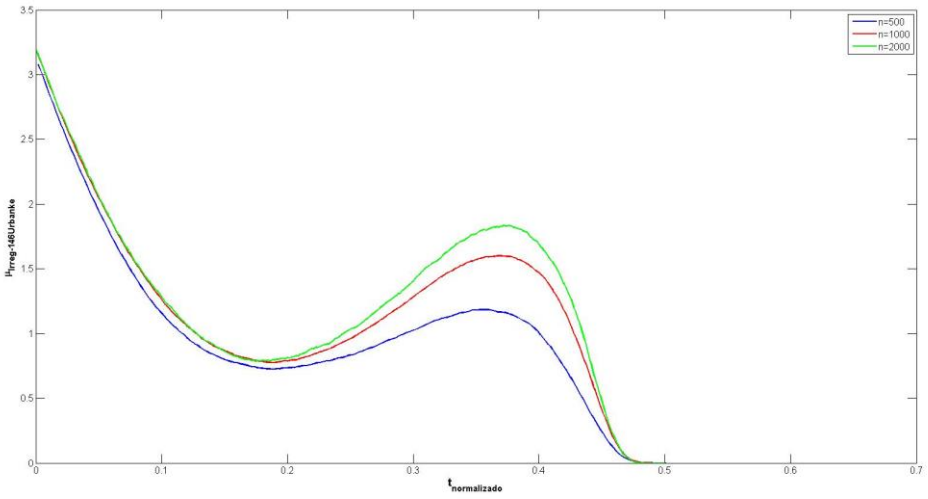

Figura 20. $\mu_{Irr-146Urb}$ para códigos de longitudes 500, 1000 y 2000

* (**Nota:** *En la figura 30, en el eje Y se representan las μ del proceso para distintas longitudes de código y en el eje X el tiempo normalizado.*)

Al igual que en los casos anteriores, en la figura 31, volvemos a observar la dependencia de $\nu_{código}$ con respecto a *n*.

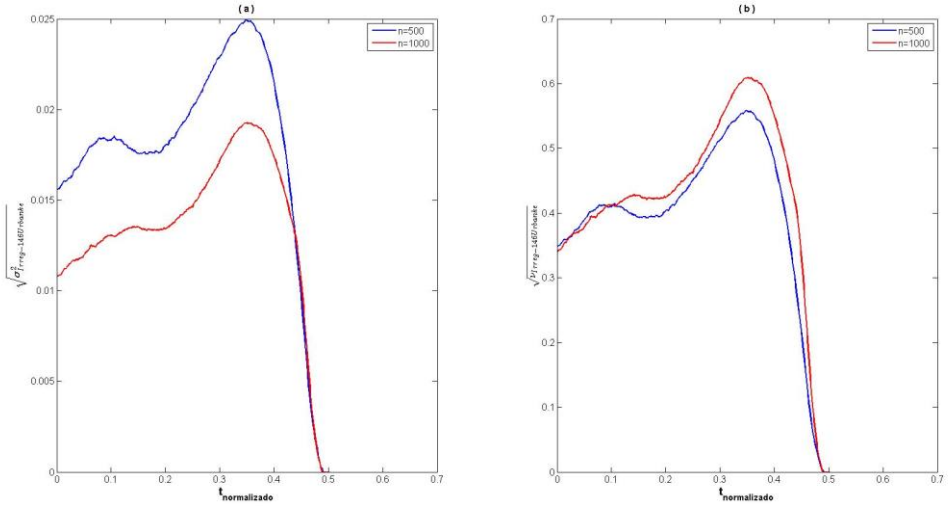

Figura 31. (a) $\sqrt{\sigma^2_{Irr-146Urb}}$ y (b) $\sqrt{\nu_{Irr-146Urb}}$ para códigos de longitudes 500 y 1000

* (**Nota:** *En la figura 31 (a), en el eje Y se representa $\sqrt{\sigma^2}$ y en el eje X el tiempo normalizado. En la figura 31 (b), en el eje Y se representa $\sqrt{\nu_{código}}$ y en el eje X el tiempo normalizado.*)

En este caso el valor de $\nu_{código}$ es:

$$\nu_{Irr-146Urb} = 0.41^2 = 0.1681$$

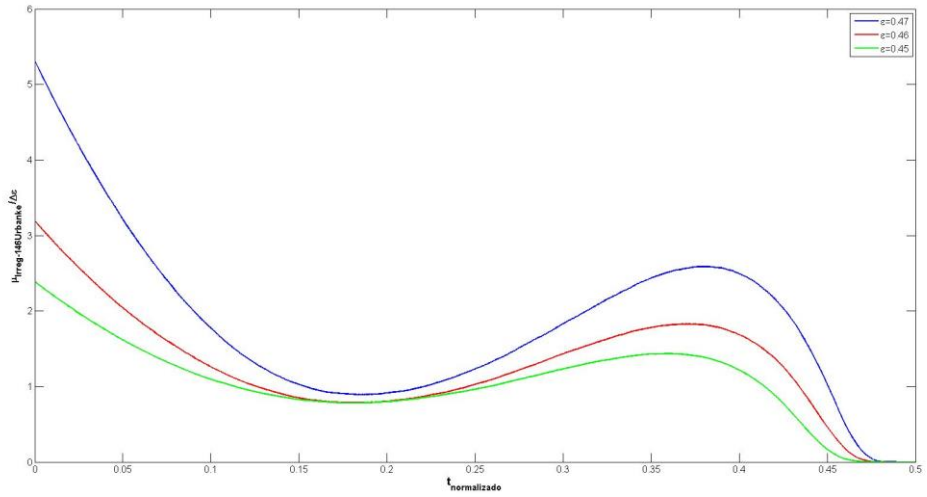

Figura 32. $\mu_{Irr-146Urb}/_{\Delta\epsilon}$ **para** $\epsilon = 0.47$, $\epsilon = 0.46$ **y** $\epsilon = 0.45$

* (**Nota:** *En la figura 32, en el eje Y se representan las* $\mu/_{\Delta\varepsilon}$ *del proceso para distintas* ε *y en el eje X el tiempo normalizado.*)

En la figura 32, se vuelve a observar la dependencia de la media de r_1 con respecto a $\Delta\epsilon$. El valor de $\gamma_{código}$ en este caso es:

$$\gamma_{Irr-146Urb} = 0.7813$$

Por último, en la figura 33, al igual que ocurría con los dos códigos anteriores, se puede ver como la dependencia de $v_{código}$ con respecto a $\Delta\epsilon$ no es lineal.

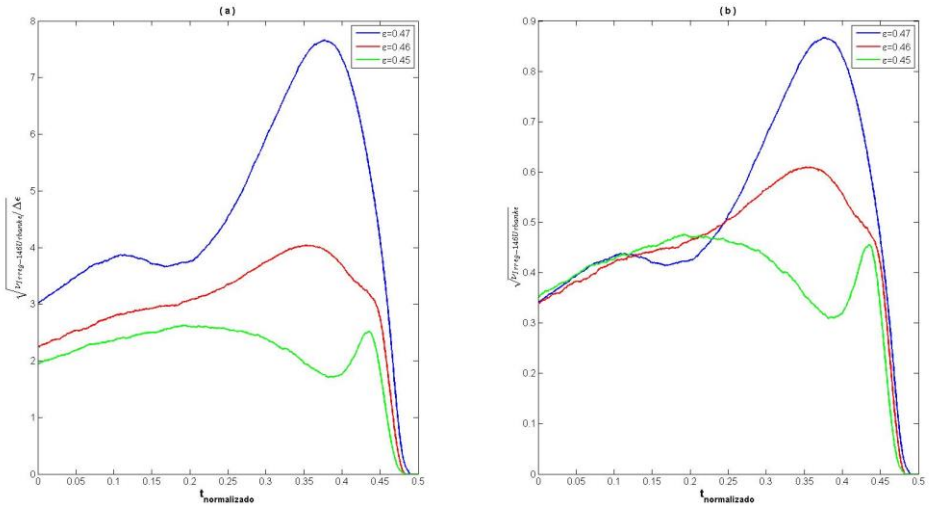

Figura 33. (a) $\sqrt{v_{Irr-146Urb}/_{\Delta\epsilon}}$ **y (b)** $\sqrt{v_{Irr-146Urb}}$ **para** $\epsilon = 0.47$, $\epsilon = 0.46$ **y** $\epsilon = 0.45$

* (**Nota:** *En la figura 33 (a), en el eje Y se representa* $\sqrt{\sigma^2/_{\Delta\varepsilon}}$ *y en el eje X el tiempo normalizado. En la figura 33 (b), en el eje Y se representa* $\sqrt{v_{código}}$ *y en el eje X el tiempo normalizado.*)

6. Estimación de la probabilidad de error para longitudes finitas

A la vista de que en todos los códigos existe un punto crítico, en torno al cual la distribución de nodos de chequeo de paridad de grado 1 se puede modelar como una gaussiana, podremos concluir que la probabilidad de error de bit es de la forma:

$$P_B \backsim \int_{-\infty}^{0} \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(u-\mu)^2}{2\sigma^2}} du \qquad (7)$$

Aplicando el siguiente cambio de variable, en la ecuación (17) se obtiene que:

$$y = \frac{u-\mu}{\sigma} \rightarrow (14) \rightarrow P_B \backsim \int_{-\infty}^{-\frac{\mu}{\sqrt{\sigma}}} N(0,1)dy \qquad (8)$$

Por lo tanto esta probabilidad de error se puede modelar mediante la función *Q* de la siguiente forma:

$$P_B \backsim Q\left(\frac{\mu}{\sqrt{\sigma^2}}\right) = Q\left(\frac{\gamma_{código} \cdot n \cdot \Delta\epsilon}{\sqrt{\frac{\nu_{código}}{n}}}\right) = Q\left(\frac{\gamma_{código} \cdot \sqrt{n} \cdot \Delta\epsilon}{\sqrt{\nu_{código}}}\right) \qquad (19)$$

Para los códigos estudiados se tienen los siguientes valores de *Q*:

$$P_{B_{Reg(3,6)}} \sim Q(2.0278 \cdot \sqrt{n} \cdot \Delta\epsilon)$$

$$P_{B_{Reg(4,8)}} \sim Q(2.041 \cdot \sqrt{n} \cdot \Delta\epsilon)$$

$$P_{B_{Irr-146Urb}} \sim Q(1.9056 \cdot \sqrt{n} \cdot \Delta\epsilon)$$

A continuación se muestran las gráficas en las que se representa la *WER* (*Word Error Rate*) para cada uno de los tres códigos que se han analizado en este estudio.

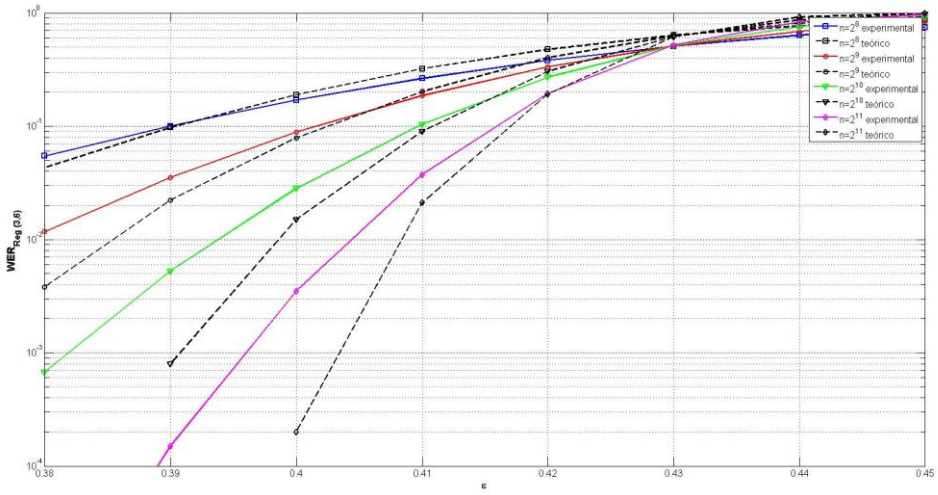

Figura 34. WER del código regular (3,6)

* (**Nota:** *En la figura 34, en el eje Y se representa la WER (línea discontinua para la teórica y línea continua para la experimental) para códigos de distintas longitudes (azul →2^8; rojo →2^9; verde →2^{10}; rosa →2^{11}) y en el eje X la ε.*)

En la figura 34, se puede ver como en el caso del código regular (3,6) las *WER* que se han obtenido de forma experimental se aproximan bastante a los resultados teóricos de estos códigos, por lo que la estimación realizada es bastante buena.

En el resto de códigos, los resultados obtenidos son muy coherentes, por lo que se puede evaluar de forma muy fiable el comportamiento de distintos códigos para n finitas.

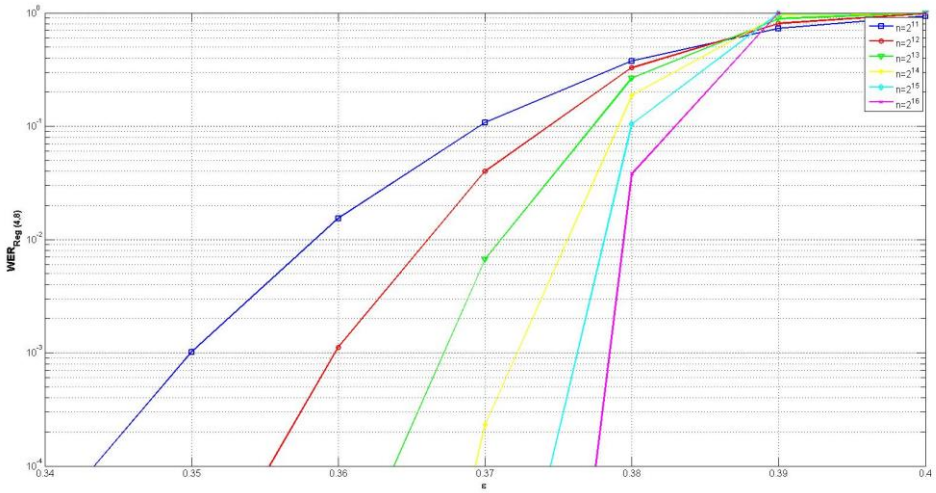

Figura 35. WER del código regular (4,8)

* (**Nota:** *En la figura 35, en el eje Y se representa la WER para códigos de distintas longitudes (azul→2^{11}; rojo→2^{12}; verde→2^{13}; amarillo→2^{14}; celeste→2^{15}; rosa→2^{16}) y en el eje X la ε.*)

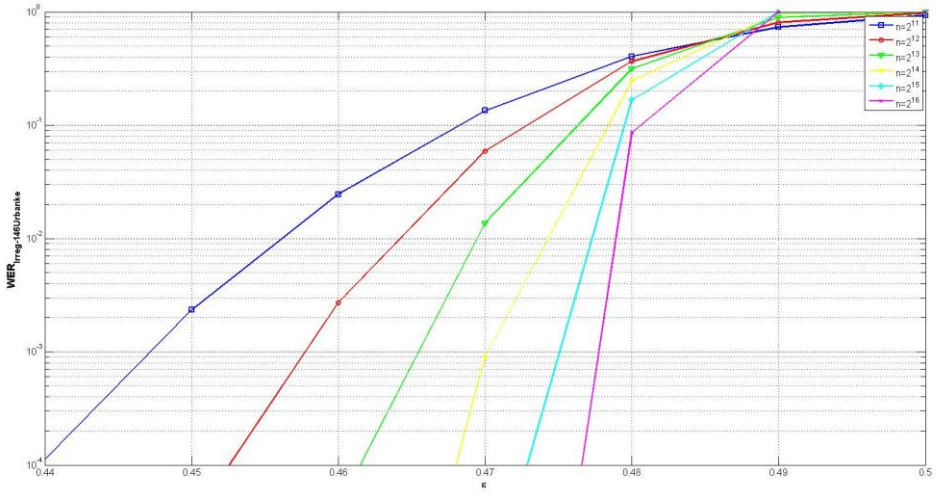

Figura 36. WER del código irregular optimizado para operar cerca de la capacidad

* (**Nota:** *En la figura 36, en el eje Y se representa la WER para códigos de distintas longitudes (azul→2^{11}; rojo→2^{12}; verde→2^{13}; amarillo→2^{14}; celeste→2^{15}; rosa→2^{16}) y en el eje X la ε.*)

En las figuras 35 y 36, se puede observar como a medida que la longitud del código $n \to \infty$, la probabilidad de error cae a 0 por debajo de $\epsilon_{TH_{código}}$.

7. Conclusiones

7.1. Importancia del software implementado y los resultados obtenidos

A pesar de que ya existen demostraciones teóricas de todo lo expuesto en este estudio [17], hemos desarrollado un software capaz de estimar de forma sencilla la ley de escalado de un código *LDPC* cualquiera.

Este software nos permite comparar distintos códigos *LPDC* para cualquier longitud de código sin necesidad de llevar a cabo costosas simulaciones.

A pesar de que la aproximación calculada mediante este software puede ser poco precisa, permite realizar estimaciones de la probabilidad de error de forma muy rápida y sin necesidad de simular códigos muy grandes.

Otra de las ventajas de este software es la posibilidad de usarlo en prácticas docentes de asignaturas de comunicaciones digitales avanzadas con el fin de ilustrar con ejemplos gráficas la teoría así facilitar al alumno su comprensión.

7.2. Líneas futuras

A pesar de que la mayor parte de los canales de comunicaciones que se usan hoy día pueden reducirse a un canal *BEC*, convendría ampliar este estudio incluyendo el canal *AWGN* (*Additive White Gaussian Noise*) e implementando como método de decodificación el algoritmo *belief propagation*, citado con anterioridad, ya que de esta manera, el canal de transmisión se asemejaría aún más a los canales reales y se reduciría considerablemente el coste computacional.

Una posible mejora basada en los resultados obtenidos de este estudio, sería la implementación de una herramienta que, dados un conjunto de códigos *LDPC* y un canal de comunicaciones, calcule los parámetros característicos de cada uno de los códigos para ese canal en concreto.

Planificación y desarrollo

Inicialmente con este proyecto se quería realizar un estudio exhaustivo del comportamiento de distintos tipos de códigos *LDPC* en diferentes canales de comunicación, pero debido a la falta de tiempo por estar realizando prácticas en una empresa de consultoría, no ha sido posible realizar tan amplio estudio, por lo que lo hemos reducido al canal *BEC* y hemos planteado como líneas futuras de trabajo el desarrollo de la herramienta que obtenga el modelo de un canal para un conjunto de códigos *LDPC*.

Presupuesto

A continuación se presenta un presupuesto del coste estimado de este proyecto.

Presupuesto del TFG

Autor	
Carlos Guzmán Velasco	
Centro	
Universidad Carlos III de Madrid	
Departamento	
Teoría de la Señal y Comunicaciones	
Título del TFG	
Estudio de códigos finitos LDPC y desarrollo de una herramienta simple de diseño	
Duración	
4 meses	

Desglose de costes del TFG

COSTE DE PERSONAL

Categoría	Salario bruto mensual	Salario neto mensual*	Dedicación	Coste
Ingeniero junior	*2000€*	*1550€*	*4 meses*	*6200€*
COSTE TOTAL				*6200€*

**Retención fiscal (IRPF) y Seguridad Social*

COSTE DE EQUIPOS

Descripción	Coste	Precio de depreciación	Dedicación	Coste de amortización
Ordenador portátil	*1000€*	*6 meses*	*60 meses*	*100€*
Licencia MATLAB	*2000€*	*6 meses*	*60 meses*	*200€*
COSTE TOTAL	*2800€*			*300€*

RESUMEN DE COSTES

Personal	*6200€*
Amortización	*300€*
COSTE TOTAL	*6500€*

Bibliografía

[1] C. E. Shannon. *"A Mathematical Theory of Communication"*. The Bell Systems Technology Journal, Vol. 27, pp. 379-423, 623-657, Julio, Ocutbre, 1948.

[2] B. Sklar. *"Digital Communications: Fundamentals and Applications"*. Prentice-Hall, Inc., New Jersey, 1988.

[3] R. G. Gallager. *"Low-Density Parity-Check Codes"*. Ph. D. Thesis, Massachusetts Institute of. Technology, Cambridge, Septiembre, Julio, 1963.

[4] D. J. C. Mackay and R. M. Neal. *"Near Shannon limit performance of low density parity check codes"*. Electronics Letters, Julio, 1996.

[5] Ian A. Glover and Peter M. Grant. *"Digital Communications, Third Edition"*. Prentice-Hall, 2010.

[6] John G. Proakis and Masoud Salehi. *"Digital Communications"*. McGraw-Hill, Enero, 2008.

[7] Antonio Artés Rodríguez y Fernando Pérez González. *"Comunicaciones Digitales"*. Pearson-Prentice Hall, Junio, 2012.

[8] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, and V. Stemann. *"Practical loss-resilient code,"* in Proc. 29th Annu. ACM Symp. Theory of Computing, 1997, pp. 150–159.

[9] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, and V. Stemann. *"Practical loss-resilient codes"*. IEEE Trans. Inform. Theory, vol.47, pp. 569–584, Febrero 2001.

[10] M. Luby,M. Mitzenmacher, A. Shokrollahi, and D. Spielman. *"Analysis of low density codes and improved designs using irregular graph"* in Proc. 30th Annu. ACM Symp. Theory of Computing, 1998, pp. 249–258.

[11] D. MacKay, S. Wilson, and M. Davey. *"Comparison of constructions of irregular Gallager codes"*. IEEE Trans. Commun., Vol. 47, pp. 1449–1454, Octubre 1999.

[12] Julián Rondón y Cecilia Sandoval. *"Diseño de un co-laboratorio remoto basado en programación modular de dispositivos VHDL aplicado a telecomunicaciones"*. Revista de la Facultad de Ingeniería Universidad Central de Venezuela, Vol.25, n.2, pp. 7-12. [Online]. Junio, 2010.

[13] ETSI. *"Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2)"*. Marzo, 2013.

[14] ETSI. *"Digital Video Broadcasting (DVB); Frame structure channel coding and modulation for a second generation digital terrestrial television broadcasting system (DVB-T2)"*. Abril, 2012.

[15] ETSI. *"Digital Video Broadcasting (DVB); Frame structure channel coding and modulation for a second generation digital transmission system for cable systems (DVB-C2)"*. Abril, 2011.

[16] Paul H. Siegel, *"An Introduction to Low-Density Parity-Check Codes"*. Electrical and Computer Engineering University of California, San Diego. Mayo, 2007.

[17] T. Richardson and R. Urbanke. "Modern Coding Theory". Cambridge University Press Octubre, 2007.

[18] A. Amraoui, A. Montanari, T. Richardson, R. Urbanke. “Finite-length scaling for iteratively decoded ldpc ensembles”. Junio, 2004.